Time of Arrival of Signal using Changes in Standard Deviation and Linear Regression Coefficients

Pete Schultz
Clarkson University
pschultz@clarkson.edu

Description

We are given a time series, which has only background noise at the beginning. The parameters of the noise are unknown. At some time t_0 a signal arrives. The quantities of interest are

- the time t_{arr}, the time of arrival of the signal.
- the peak strength of the signal
- the time t_{peak} at which the peak of the signal arrives

The algorithm is as follows:

Find `ipeak`, the time at which the signal is at peak strength. From now on consider only the subset $y_1, y_2, \ldots, y_{peak}$.

For each $1 \leq k \leq i_{peak}$, calculate the following quantities

$$\begin{aligned} m_k &= \text{slope of regression line through } (1, y_1),\ (2, y_2),\ \ldots,\ (i_{peak}, y_{peak}) \\ \sigma_k &= \text{standard deviation of } y_1, y_2, \cdots y_{peak} \\ s_k &= \sqrt{(k^2 - 1) * m_k^2 + \sigma_n^2} \end{aligned}$$

Plot the points (k, s_k) for $1, \ldots, i_{peak}$. The points should be approximately constant until the signal arrives, at which point it begins to increase. Because of randomness in the background noise and complexity of the signal, there will likely be several points at which s_n locally has this behavior. We pick a single s_n in the following way.

Compute the convex hull of the points (k, s_k); the point in question will be one of the vertices on the hull; in particular it will be one of the vertices where the convex hull is below the graph. For each pair of adjacent vertices (k_1, s_{k_1}) and (k_2, s_{k_2}) that satisfy this property, we extrapolate the line between these two points, and observe how far below (it must lie below the rest of the graph) the next vertex (k_3, s_{k_3}). We compute the *percentage* amount by which s_{k_3} lies farther above the x-axis than the line does at $x = k_3$. The percentage increase is divided by $k_3 - k_2$ to obtain a relative rate of increase in y

Do this for each vertex on the convex hull that lies below the graph. The vertex for which this relative rate of increase is a maximum, is taken to correspond to the arrival of the signal.

Mathematical Principles

The routine is based on the assumption that the time series data behaves like Gaussian noise for some period of time, until the arrival of a signal whose peak amplitude is greater than the amplitude of the background noise. It attempts to give an answer to the question of when the time series ceases to behave like Gaussian noise. We do not make any a priori assumptions on the mean or variance of the Gaussian noise.

Suppose we have a time series of data points $\{y_1, \ldots, y_n\}$, generated by Gaussian noise. For simplicity, assume that the distribution has mean 0 and variance 1. A graph of the time series should show that the values are clustered around the line $y = 0$, but are randomly above and below this line. Doing statistics on the points in the time series, we expect two things.

First, that the line of least-squares fit should be very close to $y = 0$. In particular, the slope should be close to zero. We can quantify this statement by observing that the slope

$$m = \frac{n\Sigma x_i y_i - (\Sigma x_i)(\Sigma y_i)}{n\Sigma x_i^2 - (\Sigma x_i)^2}$$

is a linear combination of the normally distributed random variables x_i. If $x_i = i$, then m is a Gaussian random variable with mean zero and standard deviation

$$\sqrt{\frac{12}{n^2-1}}.$$

Define m_n to be the slope of the least-squares line for the first n points. Then the quantity $p_n = m_n\sqrt{n^2-1}$ has normal distribution, with mean zero and standard deviation that does not depend on n. However, the plot of these values will not look like typical Gaussian noise, because the collection of m_n are not independent. Instead, they are the outcome of a random walk with smaller and smaller steps as n gets larger and larger. In this way, the plot of p_n becomes much less jagged than the plot of y_n as n increases. (figure 1).

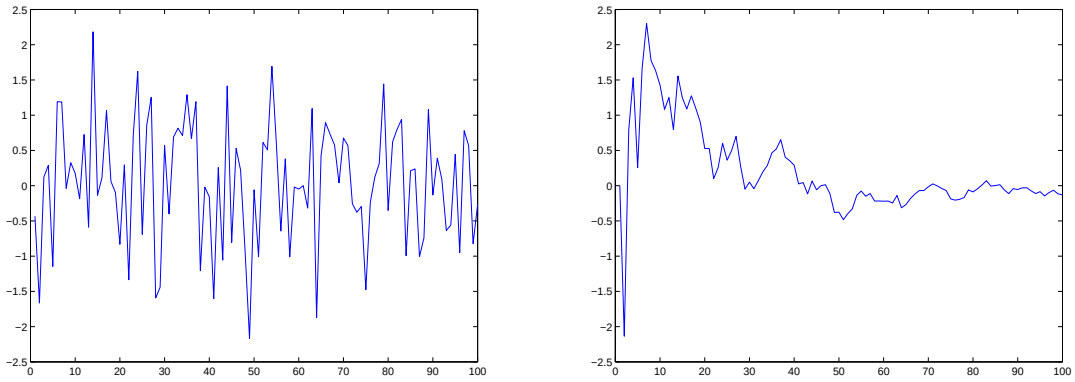

Figure 1: Comparison of the smoothness of the original data with that of p_n. Left: Gaussian noise with mean zero and unit variance. Right: Slope of p_n, as a function of n.

Second, that the sample standard deviation should be very close to $\sigma = 1$. If σ_n is the sample standard deviation of the first n points, then σ_n^2 is a random variable with mean 1. However, the σ_n are not independent for different n, so that the variation for large n will be tamer than that for y_n.

Thus we see that as long as the values of the time series are given by a Gaussian noise process, the quantities σ_n and y_n behave in a predictable manner.

What then happens when the signal arrives? Since we are assuming that the signal has greater amplitude than the typical values of the noise, we expect the standard deviation to begin increasing at the onset of the signal. Also, the presence of the signal forces several successive data points to be either all above or all below the background noise level. The result of this effect is that m_n will start to drift away from zero. We therefore expect to see that when the signal arrives, both σ_n and m_n will behave differently than they had before the signal arrived.

The algorithm works by detecting when the regression slopes and standard deviations stop behaving like those of Gaussian noise. It makes sense to consider both m_n and σ_n together. Since the signal can oscillate around the mean level of the background noise, it is quite common for m_n to oscillate back and forth between positive and negative values, and attain values close to zero for certain values of n. This tendency makes it difficult to deal with σ_n alone. On the other hand, changes in the slopes can be more readily apparent than changes is the standard deviations, so that it might be harder to pick out the signal arrival if we were only to use the standard deviations.

We therefore will work with the expression

$$s_n = \sqrt{\sigma_n^2 + p_n^2},$$

which will be referred to as the "signal detection function."

The first step in the algorithm is identify the peak signal strength. In identifying the peak signal it is important to remember that we cannot assume that the equilibrium position of the time series would be zero in the absence of noise and signal. Instead, we use the value of the first data point in the

time series as an easy approximation to the correct level, and look for the peak signal relative to y_1. The value that is returned as `peak` is the maximum over j of $|y_j - y_1|$. The first value of j that gives this maximum is the output variable `ipeak`.

If we plot the values of s_n, we expect the sequence to be relatively flat while there is only background noise, and then increase (since the standard deviation will increase, and m is expected to depart from zero.) In figure 2, there is Gaussian noise for the first 100 data points, and then a signal arrives at time 100 that is sinusoidal in shape. Note that the graph of p_n makes a sharper transition than that of the raw data.

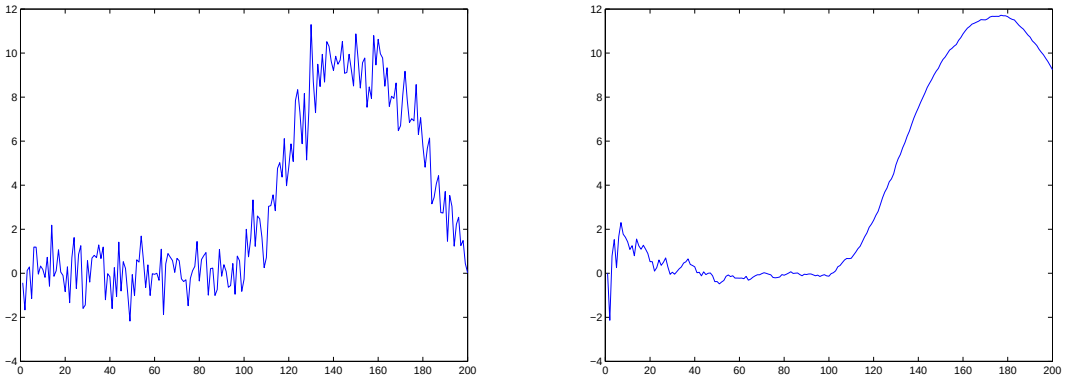

Figure 2: Comparison of the smoothness of the original data with that of p_n. Left: Gaussian noise with mean zero and variance 1. Right: Slope of p_n, as a function of n.

It is pretty clear from figure 2 where one would assign the signal arrival. The point in question is the point on the graph that "sticks out" the most. In this way we are geometrically led to consider the convex hull of the graph of the s_n's.

To decide which vertex on the convex hull we should take, we make the following observations. First, the choice of signal arrival should not depend on what units we use to measure s. Therefore s-values should be compared only to s-values, which means that we cannot consider angles or approximate curvature of the convex hull. Instead, we are forced to consider ratios between different s-values.

Suppose that (k_1, s_1), (k_2, s_2), and (k_3, s_3) are consecutive vertices, with $k_1 < k_2 < k_3$ of the convex hull of the graph of s_n, and that all three vertices lie below the graph. By convexity, the slope of the line from k_1 to k_2 must be less than that from k_2 to k_3. Extrapolate the first line to locate the point (k_3, q), where $q < s_3$. Now the s-values are all positive. At times before the signal has not yet arrived, we expect the s-values to change relatively little, so that s_3 lies very close to the extrapolated line. Once the signal arrives, the rate of increase of the s-values should pick up, so that s_3 is relatively far from the extrapolated line. We compute the relative change in s-value, $(s_3 - q)/q$. Since we expect this value to be larger if k_2 and k_3 are far apart and smaller if they are close together, we normalize by dividing by $k_3 - k_2$. This quantity is computed for each vertex on the lower part of the convex hull, and the k_2 for the largest such value is chosen as the time of arrival.

Physical and Engineering Principles

As a means of providing validation for the arrival time algorithms, the ESA team provided estimates of times of arrival for sixteen experimental datasets. These estimates were obtained by visual inspection of the graph of the time series. Figure 3 shows the results of both the ESA opinion and the arrivaltime.m algorithm. In the majority of cases, the agreement is five data points or fewer. In a couple cases the discrepancy appears quite sizable; however a visual inspection of the data in these cases shows that there is an apparent change in the data at the time identified by arrivaltime.m

For example, look at the data for SG803T13RAW, the experiment for which the discrepancy was the largest (figure 4). Both results identify a point where the data begins increasing and then oscillating

Dataset	ESA estimate	arrivaltime.m	Difference
AC002T13RAW	990	989	1
AC002T23RAW	1086	1097	-11
AC002T33RAW	1123	1116	7
AC002T43RAW	1088	1111	-23
SG801T13RAW	551	551	0
SG801T23RAW	550	555	-5
SG801T33RAW	550	550	0
SG801T43RAW	600	599	1
SG802T13RAW	504	519	-15
SG802T23RAW	513	515	-2
SG802T33RAW	503	513	-10
SG802T43RAW	570	565	5
SG803T13RAW	483	508	-25
SG803T23RAW	479	482	-3
SG803T33RAW	479	477	2
SG803T43RAW	528	528	0

Figure 3: Comparison of ESA expert opinions and arrivaltime.m for sixteen experimental datasets

with greater and greater amplitude. They differ, however, in which increase marks the signal and which is the end of the pure background noise.

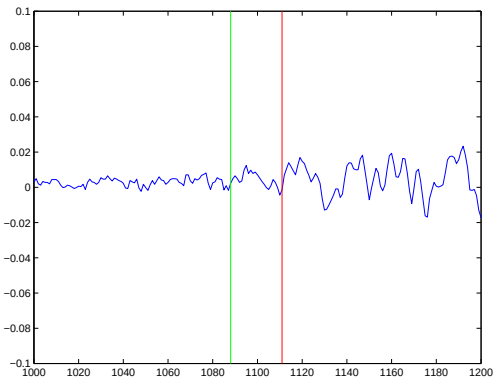

Figure 4: Accelerometer data showing different answers to the arrival time question. The left vertical line indicates the ESA opinion; the right one, the output of arrivaltime.m. The data between these marks could be interpreted as the start of the signal, or it could be interpreted as a statistical blip

The algorithm works best when the signal grows to peak amplitude at a moderate rate. If the growth is too slow, the increase in the regression slope and the standard deviation is slow enough that it is difficult to see. If the growth is too rapid too early, the routine of picking out the greatest increase will tend to pick the point of maximum growth instead of the onset of the signal. This explains the discrepancy for SG803T13RAW (figure 5), the dataset that gave the biggest discrepancy of 25 datapoints. Figure 6 shows a portion of the graph of the signal detection function and its convex hull. The prime choices for the signal onset are at $t = 484$ and at $t = 508$. The former would give excellent agreement with the ESA opinion; however due to the rapid increase in the signal beginning at $t \approx 503$, the algorithm finds $t = 508$ to be more significant.

Usage

The matlab command arrivaltime has the following syntax:

```
[iarr,peak,ipeak,out] = arrivaltime(dvec,doplot)
```

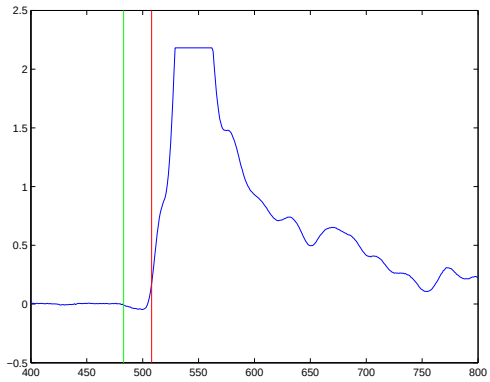

Figure 5: Strain gauge data showing the effect of early quick growth of the signal amplitude. The left vertical line indicates the ESA opinion; the right vertical line is the output of arrivaltime.m Here the rapid change at $t = 508$ causes the algorithm to interpret any prior variations in the data as insignificant.

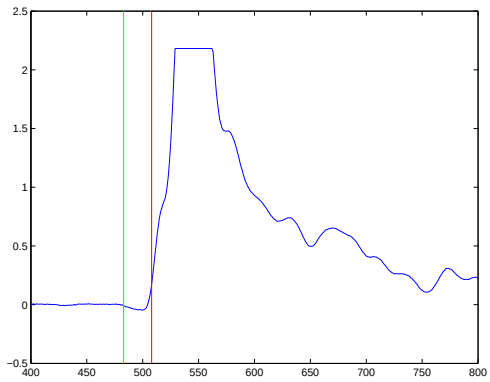

Figure 6: Signal detection function and convex hull for data set SG803T13RAW. The candidates for signal onset are marked by vertical lines.

```
INPUT:
    dvec      the vector containing the time series
    doplot    a string that indicates whether to draw plots.  Setting doplot to
                   ‘plot‘ draws illustrative plots.  If doplot is anything else, do
                   not draw plots.  The default is not to draw plots.

OUTPUT:
    iarr      the index into dvec corresponding to the time the signal arrived.
    peak      the largest amplitude of the signal
    ipeak     the index into dvec corresponding to the time the peak of the signal
                   arrived.
    out       an ipeak-by-3 matrix.
                   Column 1 contains standard deviations
                   Column 2 contains slopes of the linear regression lines
                   Column 3 contains the signal detection function
```

At present, the algorithm has no parameters that affect the output. The parameter ‘doplot’ controls whether graphics are produced. If ‘doplot‘ is set to ‘plot‘, the routine produces two figures, generated in figure 1 and figure 2.

Figure 1 draws the time series. Figure 2 plots the signal detection function (see the section Mathematical Principles) and its convex hull. This figure illustrates which times were considered as candidates for the time of arrival. It can show the user whether there are any close misses.